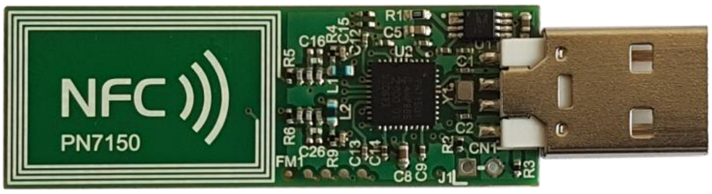
NFC NXP-NCI USB dongle
Linux quick start guide

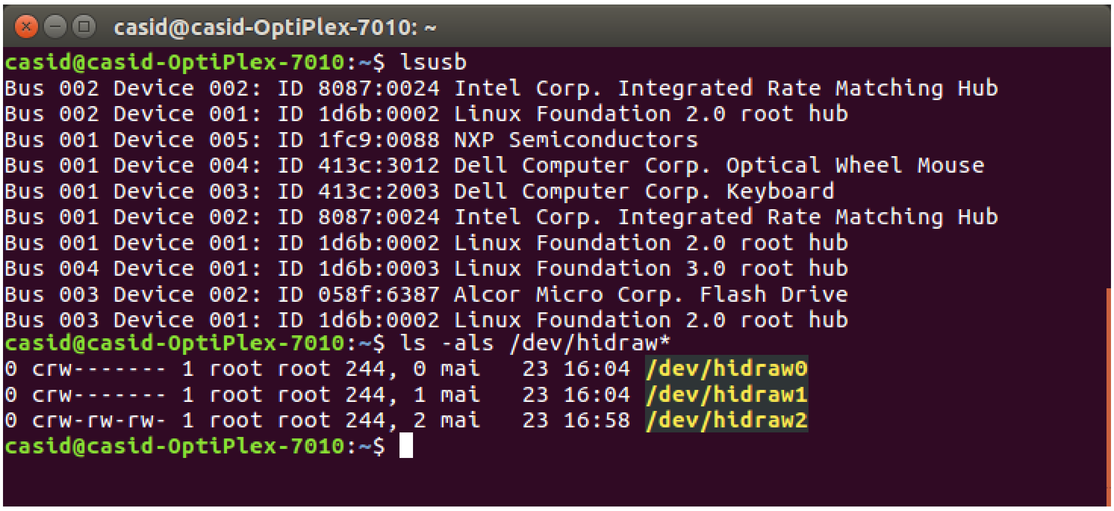


General comments

- The NFC functionality is insured thanks to NXP PN7150 NFC Controller
- USB interfaced is provided via HID standard thanks to NXP LPC11u24 MCU
- The dongle is supported under Linux based devices through generic HID support of this operating system
- Example is provided to demonstrate full P2P functionality (reader, P2P and card emulation) of the NFC dongle

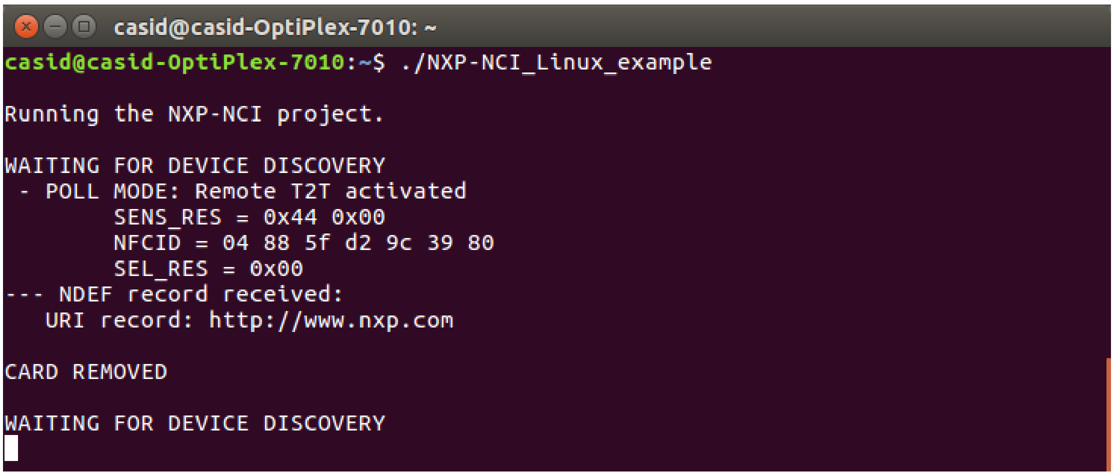
Quick start

- First of all permission of the dongle shall be set in Udev: Just drop the provided file named “99-hid.rules” (optionally rename it) to “/etc/udev/rules.d” directory of the target before plugging the dongle in the USB port.
- The dongle is detected as “LPC I2C HID” in the USB devices list, and permission to r+w are set:



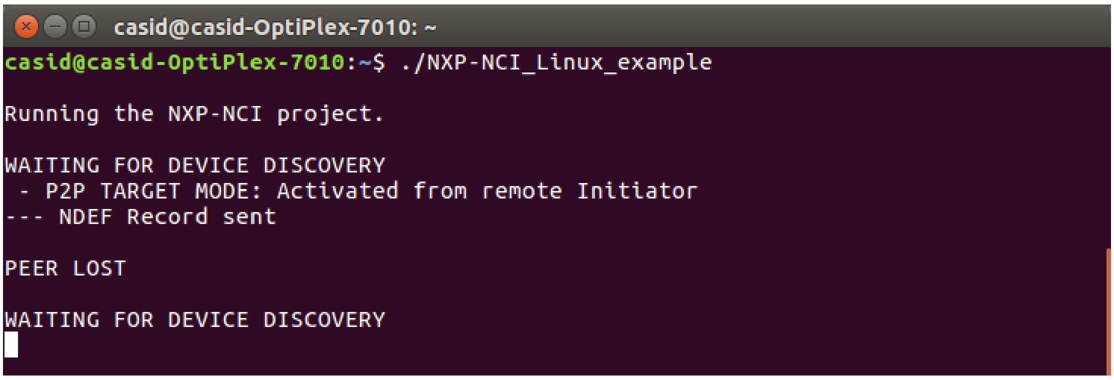
```
casid@casid-OptiPlex-7010:~$ lsusb
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 005: ID 1fc9:0088 NXP Semiconductors
Bus 001 Device 004: ID 413c:3012 Dell Computer Corp. Optical Wheel Mouse
Bus 001 Device 003: ID 413c:2003 Dell Computer Corp. Keyboard
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 002: ID 058f:6387 Alcor Micro Corp. Flash Drive
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
casid@casid-OptiPlex-7010:~$ ls -als /dev/hidraw*
0 crw------- 1 root root 244, 0 mai   23 16:04 /dev/hidraw0
0 crw------- 1 root root 244, 1 mai   23 16:04 /dev/hidraw1
0 crw-rw-rw- 1 root root 244, 2 mai   23 16:58 /dev/hidraw2
casid@casid-OptiPlex-7010:~$ 
```

- In a Terminal window, running provided “NXP-NCI_Linux_example” executable launches NFC discovery. Tapping a card generates such display:



```
casid@casid-OptiPlex-7010:~$ ./NXP-NCI_Linux_example

Running the NXP-NCI project.

WAITING FOR DEVICE DISCOVERY
 - POLL MODE: Remote T2T activated
        SENS_RES = 0x44 0x00
        NFCID = 04 88 5f d2 9c 39 80
        SEL_RES = 0x00
--- NDEF record received:
   URI record: http://www.nxp.com

CARD REMOVED

WAITING FOR DEVICE DISCOVERY
```

- While tapping a NFC phone triggers the reception of NDEF text record on the phone and the following log:



```
casid@casid-OptiPlex-7010:~$ ./NXP-NCI_Linux_example

Running the NXP-NCI project.

WAITING FOR DEVICE DISCOVERY
 - P2P TARGET MODE: Activated from remote Initiator
--- NDEF Record sent

PEER LOST

WAITING FOR DEVICE DISCOVERY
```

Building and debugging the example

- Make sure the following components are installed:
    - gcc / g++ (e.g. “apt-get install gcc g++)
    - gdb (e.g. “apt-get install gdb)
    - Eclipse CDT (e.g. “apt-get install eclipse-cdt)
    - Udev (e.g. “apt-get install libudev-dev)
    - pthread (e.g. “apt-get install libpthread-stubs0-dev)
- Example source code is delivered in the form of Eclipse C/C++ project
- The project must be imported into Eclipse C/C++ IDE:
    - File->Import…->Existing projects into Workspace->Select archive file
    - Select the NXP-NCI_example.zip package
    - Tick the “NXP-NCI_Linux_example” project and click Finish button
- Build the project in “Debug” mode
- Then start “Debug As -> Local C/C++ Application”
- Note that the Debug configuration traces all NCI exchanges in the console (enabled according to NCI_DEBUG definition inside GCC C Compiler Preprocessor defined symbols of the project properties)